10 Predictions for Developers in 2019

By Siddhartha Agarwal

developer.oracle.com

ORACLE®

1 Eight out of 10 custom applications will include embedded intelligence as a first-class capability by 2020.

So developers need to understand how data science works. Developers aren't going to become data scientists — one writes code, one thinks in math and models. But devs will increasingly need to understand data science methodologies, and to integrate data science models into their workflow. Data makes software more intelligent, giving it the ability to predict outcomes or anticipate user needs through machine learning. So developers increasingly need a new level of partnership with data scientists to make great work happen. Developers can expose the data scientists' models via APIs, and embed them in domain-specific apps to really drive change. Consider a retailer trying to intelligently decide which brick and mortar store to fulfill e-commerce orders from. A data scientist can create the model that calculates the optimal store from which to ship, so that the company ships the sweater that was likely to sit on the shelf at a store in a warm location rather than the one likely to get bought by an in-store shopper in frigid areas. A developer could pull that kind of intelligence into a fulfillment app, and put it in employees' hands to make the right decision.

2 Developers will need to partner with a platform engineer, which will emerge as a key new role for cloud native development.

Think of this new platform engineer role as something like the sys admin for the cloud. The platform engineer will marshal all the resources needed to have a high performing cloud infrastructure on which developers can build and deploy their cloud native applications. Platform engineers will understand elements such as how to employ kubernetes to orchestrate containers, and how to use tools like Istio and Envoy proxies to run a service mesh, which is increasingly important for the complex service-to-service communication of microservice applications. They'll also focus on ensuring security of containers, and of communications between services, since both security and availability are essential to using containers at scale within the enterprise. Developers still get the benefits of the cloud — they can develop and deploy in containers, and the cloud will scale resources up and down to serve their apps based on demand. But it's a platform engineer who will pull all that infrastructure together and have it ready for developers to use.

3 To balance security and performance needs, a hybrid model that falls between virtual machines and containers will rise in popularity for deploying applications.

The past year brought some nasty security threats that should again remind developers that nothing else matters if they get security wrong. In response, look for an application deployment approach this coming year that delivers most of the performance and efficiency of running in containers, but in a super lightweight virtual machine that assures better security and isolation in multi-tenant environments. Kata Containers offer one promising implementation of this idea, but similar concepts will arise elsewhere. Cloud service providers will drive this innovation, but developers need to understand these options and what performance tradeoffs and security gains they can bring.

4 The economics of serverless become too compelling to ignore, driving serverless innovation on multiple fronts.

For starters, an open standard for serverless will take clear shape, allowing developers to embrace serverless like they have kubernetes, without the lock-in risk many of today's serverless options have. We're not predicting exactly how that will take shape — one leading platform, or several platforms with common, standards-based building blocks? — but we'll see major progress by this time next year. Second, serverless solutions will rapidly extend from today's function-oriented solutions (Oracle Fn, AWS Lambda, Azure Functions) to other business domains, such as cloud-first, serverless infrastructure for large-scale data processing based around the Apache Spark, Flink, and Kafka ecosystems. Like running big data on the cloud, the economics are just too compelling. Overall, look for modern, open cloud architectures to rely heavily on serverless the coming year as a way to abstract away additional platform dependencies, accelerating open cloud development.

5 As bots proliferate, developers turn to 'assistants' to find the right bot for the job.

Developers are going to write lots of bots, with each one representing some kind of skill -- taking an order, looking up inventory, or scheduling a delivery. But the user can't be expected to hunt down the right bot for every job. Instead, users need an intelligent interface — a digital assistant — that can look at each possible bot, and recruit it as needed to get a task done. So if you go into your HR app, you can tell a digital assistant that you want to file expense reports, and that assistant will call upon the bot automation to execute your filing. Conversational interfaces will keep growing in intelligence, and the un-intelligent conversational interfaces will fade away. For certain use cases, like order taking or inventory look ups mentioned earlier, intelligent interfaces will provide more value than human interactions.

6 Developers choose their clouds based on openness, and reject cloud lock in:

Developers move to cloud infrastructure so they can focus on creating, rather than on IT operations. So when they pick a cloud to build on, they want a choice of languages, databases, and compute shapes. And they want to be able to move workloads to any cloud.They're increasingly wary of forked open source that limits the ability to move among clouds, or the ability to adopt new innovations in the unforked branches. That's why open source and open standards will take a center role as developers choose clouds that avoid cloud lock-in.

7 Developers decide one cloud isn't enough.

While the first wave of cloud often leaned on a single provider, the current expansion will be a multi cloud strategy, especially with the plethora of SaaS applications available, spreading workloads across multiple providers. The move to cloud is a staged journey, and many on-premises workloads will remain for years to come.

Developers will demand the same tools and workflow across cloud and on-premises deployments. An API that today pulls data from an on-prem HR app might next year pull that same data from a software-as-a-service HR app, routing the data to a customer application running on a public cloud.

8 Autonomous Databases will let developers speed their applications to market scale:

Developers want more than rapid prototyping. They also want, when that prototype hits the mark, to get as many people using their work, as fast as they can. That scale validates success -- and it lets developers get the feedback to make an application better. A cloud-based, autonomous database makes both speed and scale possible. An autonomous database means a developer doesn't have to think about scaling, patching, provisioning, or tuning a database. Instead, they can just focus on building their app, knowing they have a repository for their data that will automatically scale to meet their needs. Developers gain productivity since they can launch a high-powered database in minutes without help from a DBA. Scale comes from the cloud, letting a business ramp up as much capacity as needed, and to run it efficiently by letting the database tune and secure itself.

9 Legacy, enterprise applications jump to cloud-native development approaches:

By now it's clear that cloud native is the default choice for new, in-house development projects. The harder part, though, is that 80%-plus of your tech stack that's legacy and not cloud native. Look for new technologies and tactics that make it possible to modernize and containerize legacy, enterprise apps, so you can deploy, run and manage them in much the same way as cloud-native apps. Two specific examples of bringing cloud-native approaches to enterprise apps: new operators that can stand up WebLogic domains in Kubernetes, and Helidon, a framework for building Java applications as microservices to modernize Java monoliths. This trend is good news for experienced developers, since it lets them build on the skills they have and extend them into container-based, cloud-native environments.

10 If you're a developer in a regulated industry, blockchain is coming your way.

Developers working on applications that draw data from external sources will soon be querying some of that data from a blockchain ledger, especially if the applications are used in regulated industries such as food and agriculture, pharmaceuticals, or finance. Developers of cloud applications now can embed features that call out to a blockchain cloud service, outside of their app. Using blockchain as a cloud service means developers don't have to worry about the complexity of implementing blockchain technology, such as assuring the performance and scalability. Why blockchain, and not a database? Blockchain can deliver greater trust and transparency around data coming from third parties such as suppliers and shippers, and that trusted data can make it easier to comply with regulations.